

Homework 1 - Movimiento Browniano

Claudia Lizeth Hernández Ramírez

31 de agosto de 2021

Resumen

LA distancia es directamente proporcional a los parámetros n_{800} , n_{8000} , n_{80000} (largo de caminata) y a dimensión.

1. Introducción

Este trabajo tiene como objetivo analizar y comprender el efecto que provoca modificar los parámetros de la dimensión y el largo de caminata sobre la distancia que recorre una partícula. Movimiento Browniano.

2. Desarrollo

Comencé por determinar el recorrido de mi partícula modificando en el programa (figura 1) [8] los parámetros DIM y DUR que representan las dimensiones (3, 6, 9, 27, 81 y 243) y el largo que durará la caminata (800, 8000 y 80000 pasos) respectivamente; estos datos los capturé en un archivo de excel (figura 2) [2]. Para comenzar a graficar se utilizó la librería `ggplot2`. [6] [1] Se importó el archivo de excel antes creado DATAHW1 y con `geom-boxplot` [9] generé las primeras cajas para $n=80000$; sin embargo Rstudio acomoda las cajas en un orden que no era entendible (figura 3) [3]. Con `fct-relevel` [7] logré ordenar de forma manual el eje X para que pudiera ser entendible (figura 4). Para lograr que n_{80000} , n_{8000} y n_{800} [5] [4] estuvieran en un mismo gráfico utilice el símbolo de suma (+) posterior a este símbolo coloqué de nuevo un `geom-boxplot` pero con eje X de n_{8000} , posteriormente otro `geom-boxplot` con eje de n_{800} (figura 5). Con la librería `scales` añadí el título principal de la gráfica, el subtítulo y los nombres de los ejes X y Y. Finalmente este es el código (figura 6) con el cual pude obtener la gráfica (figura 7) que describía el comportamiento del eje `distancia` con la variación de los parámetros antes mencionados.

3. Conclusión

La distancia que se recorre conforme se aumenta de dimensiones se muestra en un comportamiento ascendente, este patrón se replica para $n=800$, $n=8000$ y $n=80000$, es decir, la distancia es directamente proporcional al número de dimensiones. De igual forma, si analizamos una sola dimensión, por ejemplo D27, es evidente que la distancia mayor recorrida es mayor conforme aumentamos el largo de caminata.

Al momento de trabajar. Tuve un par de dificultades para realizar esta tarea, al inicio pretendía colocar todos los datos obtenidos de la caminata en forma de vector en Rstudio, sin embargo no logré acomodar la gráfica de manera correcta, en la figura 8 se puede observar como se graficaba. Posteriormente en la librería `ggplot` ví que era mucho más sencillo trabajar las gráficas, pero seguía teniendo el mismo problema de no saber como sobreponer las gráficas así que, se me ocurrió generar un boxplot para cada n y después juntarlas ya que había visto que se podía juntar una gráfica boxplot con una de puntos y pensé que se podía hacer lo mismo con dos gráficos similares, lo intenté y funcionó, el resto fue cuestión solamente de dar formato. Personalmente creo que la escala logarítmica del eje Y no está del todo clara ya que las boxplot de la D81 y D243 no se logran visualizar abiertamente, de ahí en fuera estoy contenta con mi trabajo ya que personalmente esta materia se me complica.

```

1 dim = 243
2 dur = 80000
3 pos = rep(0, dim)
4 mayor = 0
5 for (t in 1:dur) {
6   cual = sample(1:dim, 1)
7   if (runif(1) < 0.5) {
8     pos[cual] = pos[cual] + 1
9   } else {
10    pos[cual] = pos[cual] - 1
11  }
12  mayor = max(mayor, sum(abs(pos))) #manhattan
13 }
14 print(mayor)

```

Figura 1: Programa utilizado para determinar la distancia recorrida.

dimension	n800	n8000	n80000
D1	39	85	747
D1	25	160	215
D1	32	97	308
D1	30	84	396
D1	52	37	378
D1	35	127	609
D1	38	109	465
D1	46	84	288
D1	54	130	271
D1	12	39	309
D3	39	114	548
D3	48	151	569

Figura 2: Fragmento de datos capturados de caminata en excel - DATAHW1

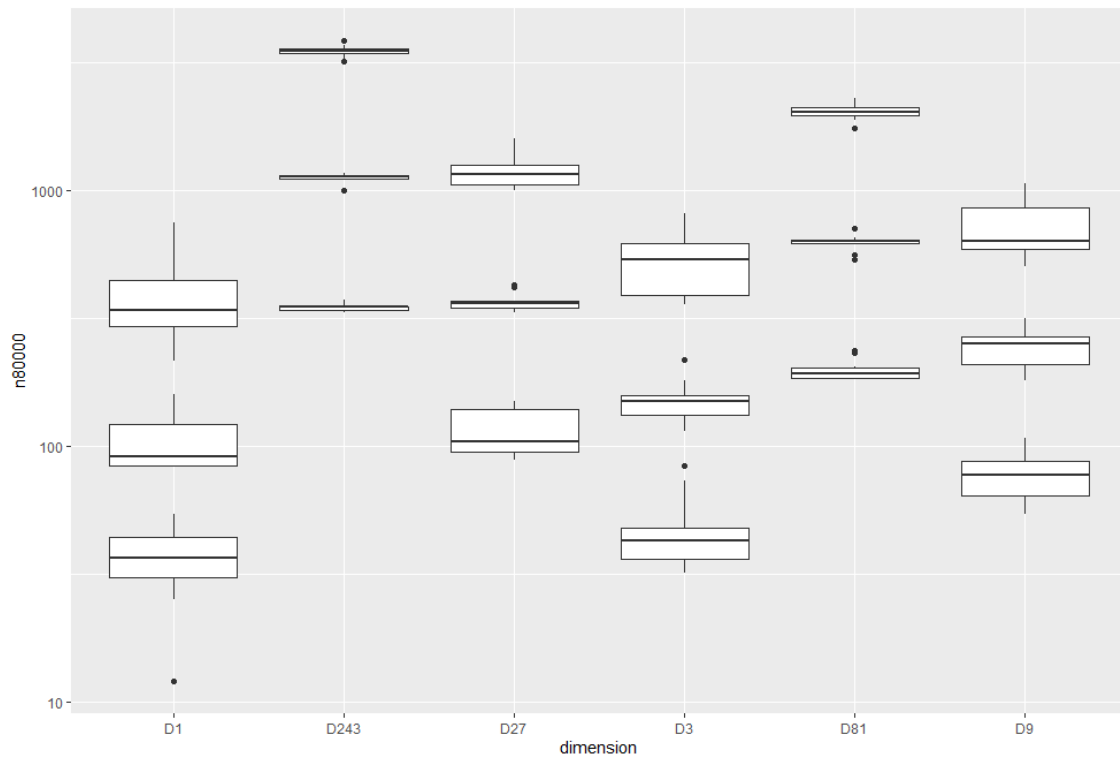


Figura 3: Primer intento de gráfica, eje X desordenado.

```
geom_boxplot(mapping = aes(x = fct_relevel(dimension, "D1", "D3", "D9", "D27", "D81", "D243"))).
```

Figura 4: Implementación de fct-relevel para reordenar manualmente eje X.

```
ggplot(data = DATAHw1)+
  geom_boxplot(mapping = aes(x = fct_relevel(dimension, "D1", "D3", "D9", "D27", "D81", "D243"), y = n80000, fill = "n80000"))+
  geom_boxplot(mapping = aes(x = dimension, y = n8000, fill = "n8000"))+
  geom_boxplot(mapping = aes(x = dimension, y = n800, fill = "n800")) +
  scale_y_log10() +
```

Figura 5: geom-boxplot + geom-boxplot + geom-boxplot.

```
#intento para reordenar eje X
ggplot(data = DATAHw1)+
  geom_boxplot(mapping = aes(x = fct_relevel(dimension, "D1", "D3", "D9", "D27", "D81", "D243"),
    y = n80000, fill = "n80000"))+
  geom_boxplot(mapping = aes(x = dimension, y = n8000, fill = "n8000"))+
  geom_boxplot(mapping = aes(x = dimension, y = n800, fill = "n800")) +
  scale_y_log10() +
  labs(title = "Efecto de la variación de dimensiones y largo de caminatas sobre la distancia",
    subtitle = "Distancia Manhattan", x = "Dimensión", y = "Distancia")
```

Figura 6: código final para generar BOXPLOT final.

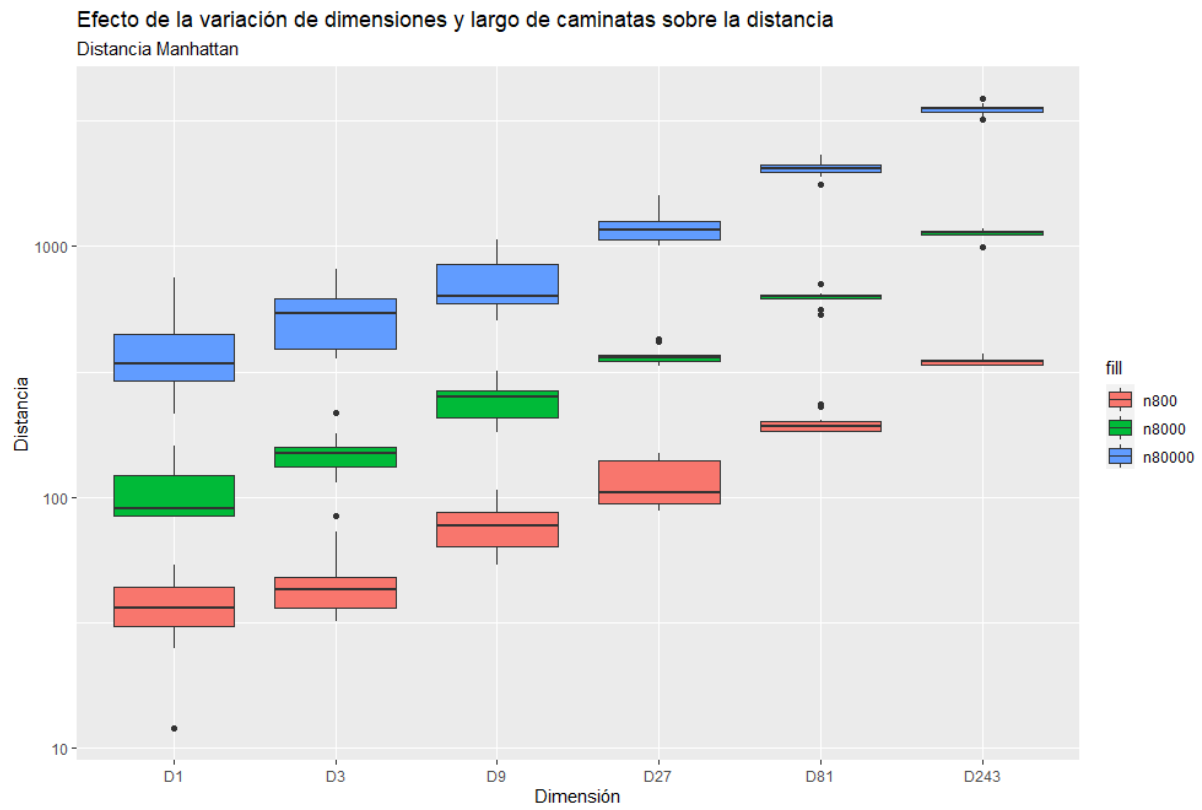


Figura 7: BOXPLOT final.

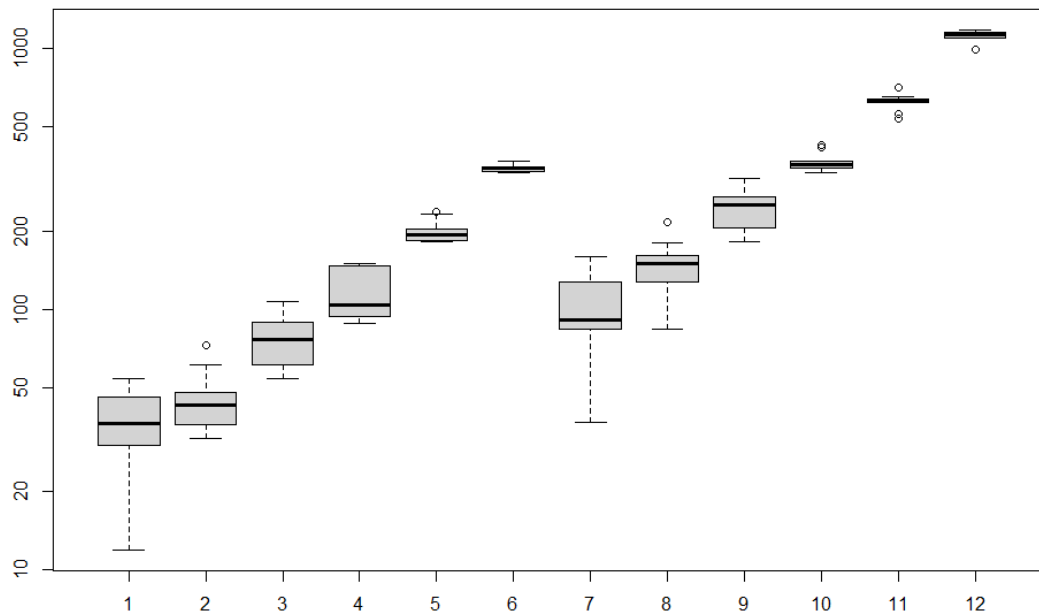


Figura 8: Primer intento de boxplot.

Referencias

- [1] 4fit. Como hacer histogramas y boxplots en R con ggplot. <https://www.youtube.com/watch?v=6vKYZsKpeWg/>, 2020. [Online; accessed 27-August-2020].
- [2] Alium calidad y productividad. Gráfica de Caja en Microsoft Excel. <https://www.youtube.com/watch?v=F-z0CZho4RI/>, 2017. [Online; accessed 27-August-2020].
- [3] Equitable Equations. Boxplots in R. <https://www.youtube.com/watch?v=ETbFdSvPwUw&t=287s/>, 2020. [Online; accessed 28-August-2020].
- [4] Rafael González. Cómo SUPERPONER varias GRÁFICAS [ggplot + R y RStudio . <https://www.youtube.com/watch?v=259PhB2pnzo/>, 2019. [Online; accessed 29-August-2020].
- [5] Rafael González. Cómo superponer varias gráficas en ggplot con r y rstudio. *RGG*, 0(0):0, 2020. doi: 0.0/.
- [6] Javier Martínez. Introducción a la visualización de datos en r con ggplot2. *NA*, 0(0):0, 2016. doi: 0.0/.
- [7] Martín Paladino. Gráficos en r con ggplot2. *NA*, 0(0):0, 2020. doi: 0.0/.
- [8] Dra Elisa Schaeffer. Repositorio github, simulation, brownian motion. *GitHub*, 0(0):0, 2020. doi: 0.0/.
- [9] Unkwnown. Boxplot en ggplot2 a partir de un vector. *RCoder*, 0(0):0, 2020. doi: 0.0/.